receiving task information from at least one of an application coordinator and the more than one computing node;

maintaining an activity scheduler list relating to available processes at said computing nodes and an activity priority list based on said task information.

REMARKS

The Examiner has rejected pending Claims 1 and 11 under 35 USC § 103 as unpatentable over the combined teachings of the Zolnowsky patent, the Custer reference and the admitted prior art (AAPA) regarding the AIX technology. In addition, the Examiner has rejected Claims 1, 11, 12 and 14 under 35 USC § 103 as unpatentable over the teachings of the Boland patent in combination with the teachings of the Custer reference and the AAPA. Claims 2, 4-10, 14-16 and 18 stand rejected as unpatentable over Boland in view of Custer and the AAPA and further in view of the teachings of the Cameron patent. Finally, Claims 3 and 17 have been rejected as unpatentable over Boland in view of Custer, the AAPA, and

Cameron. By this amendment, Claim 15 has been canceled, Claims 19 and 20 have been added, and Claims 1, 11 and 16 have been amended. No new matter is introduced by the amendments and no additional filing fee is believed due.

Applicants appreciate that the Examiner has considered the previously submitted arguments with regard to the references which were, and are again, cited against the application and has effectively withdrawn those rejections. Applicants maintain, however, that the cited prior art does not obviate the invention as claimed. Applicants further reiterate the previously-submitted request that the Applicants' arguments be explicitly addresses so that the Applicants can understand fully the basis for the Examiner's conclusions in order to more completely respond to same, particularly if the same references are again used in rejecting the claims.

The present invention provides a system and method wherein a local scheduler maintains a local priority list of ready-to-execute tasks correlated with local processes, which list is updated in accordance with a global prioritized schedule provided from the global scheduler. As encompassed by the independent claims, and as set forth

explicitly in the dependent claims, the local prioritized list may be updated by the local scheduler or by the global scheduler. As stated previously, the present approach of global (including inter-node) and local scheduling minimizes unused CPU time when an individual task is temporarily blocked or suspended waiting for I/O. Furthermore, the task used to fill idle time under the present invention is the next highest task in importance. While some prior art approaches do attempt to interject higher priority tasks for execution at local processors, none of the cited prior art, alone or in combination with each other, provides the apparatus or the method as set forth in the amended claims, including the means and the steps for dynamically creating a global prioritized schedule of said plurality of tasks, said schedule including tasks of said more than one application; communicating said global prioritized schedule to said more than one computing node; determining correspondence between said plurality of tasks and said plurality of local processes; and dynamically prioritizing said local processes in accordance with said global prioritized schedule to allow simultaneous execution of tasks from said more than one application.

The Examiner first cites the Zolnowsky patent as the primary reference against Claims 1 and 11, in combination with the teachings of Custer and the AAPA. The Zolnowsky patent provides a dispatcher model which maintains a global dispatch queue for non-bound higher priority real time threads. Each processor in the multiprocessor environment selects a candidate thread from its own queue, compares the selected thread with threads in the global real-time queue, chooses the higher priority thread for execution, and then performs a synchronization algorithm to verify that no other processor has already elected to execute the thread from the global real time queue. The Zolnowsky patent does not provide any teachings with regard to a global scheduler which creates a global prioritized scheduler and communicates that schedule to local computing nodes at which the local prioritized schedule is updated.

Under the Zolnowsky patent, there is one scheduler for each one processor, each of which has an associated dispatch queue. In the Zolnowsky system, a processor can select a thread from the global queue or its own queue. As mentioned above, when a processor selects a thread, it performs a synchronization algorithm to ensure that another processor

does not also select that thread. The Zolnowsky system has a "high priority real time queue" on which any of the processors can place a thread (see: column 7, lines 43-48). The high priority real time queue is not a global scheduler means which dynamically creates a global prioritized schedule which is then used by local schedulers. In addition, the Zolnowsky patent does not provide at least one local scheduler associated with each the more than one computing nodes which each have a plurality of processes. In Zolnowsky, there is one scheduler per processor. Furthermore, the schedulers (i.e., the one scheduler per processor) do not adhere to a prioritized schedule obtained from a global scheduler, but rather "...determine when and which threads are to be dispatched for execution on the system processors" (see: column 7, lines 15-20) in accordance with a protocol which is schematically illustrated in Zolnowsky's Figure 8. Clearly the Zolnowsky patent does not provide any teaching or suggestion of the claimed system or method wherein both global and local schedulers are available and wherein a global prioritized schedule is created, updated, and communicated to the local schedulers for updating a local prioritized list.

The Examiner has additionally cited the Custer and AAPA references in combination with the Zolnowsky patent teachings. Applicants respectfully assert that the Custer teachings with regard to threads and processes provide nothing which would motivate someone having skill in the relevant art to create a system including global and local schedulers wherein a global prioritized list is provided to the local scheduler for updating the local prioritized list.

The Examiner has cited Custer page 84 as teaching the combining of a thread with a process. The Examiner further avers that the Zolnowsky dispatching of threads together with the Custer combining of threads to processes obviates the means for "ascertaining which process(es) are assigned to the tasks". It is respectfully asserted that the Custer disclosure regarding threads and processes does not teach or suggest ascertaining which tasks are to be assigned to which processes by a local scheduler in implementing a prioritized schedule from a global scheduler. Even if one were to combine the teachings of Zolnowsky and Custer, one would not arrive at the invention as claimed. In combining Zolnowsky and Custer, one would simply have the scheduler which is associated with the one processor identifying which entity

within the process is to execute. Custer does not supply any teachings which could suggest modifying Zolnowsky to provide for ascertaining which tasks are assigned to which processes and further for prioritizing in accordance with a global prioritization schedule.

The Examiner goes on to acknowledge that the combination of Zolnowsky and Custer would not teach means for prioritizing the processes according to the prioritized schedule. Applicants respectfully contend that the AAPA would not logically be combined with Zolnowsky and Custer, and further that any such combination would not yield the invention as presently claimed. It is herein reiterated that the AIX teachings from the AAPA relate to a system wherein only one application is active at a time and wherein the assignment of priorities of processes is simply a resource utilization mechanism. There is no suggestion in the AIX prior art that process priorities would or could be used for the purpose of anticipatory scheduling of multiple tasks of different applications to be executed simultaneously. The only applicability of the AAPA teachings to the Zolnowsky system would be to replace the Figure 8 protocol, which one of ordinary skill in the art

would not be inclined to do, as Applicants had previously argued, because it would limit the functioning of the Zolnowsky scheduler. In light of the lack of teaching or suggestion of the claim features by any of the cited references, Applicants respectfully aver that Claims 1 and 11 are patentable over the combined teachings of the Zolnowsky, Custer and AAPA references.

The Examiner has additionally rejected Claims 1 and 11, along with Claims 12 and 14 as unpatentable over the combined teachings of Boland, Custer and the AAPA. The bulk of the teachings of the Boland patent relate to an affinity-based distribution of work in a multiprocessor environment having one node and one scheduler. The single scheduler 22 looks at the global run queue 24 to determine if the next waiting process has affinity with a processor. Based on the determination, the global priority run queue 26 may be updated and a selected process is sent to the processor. Under those teachings of Boland, it is clear that there is no multinode environment, no local scheduler, no local scheduler prioritized list, no communication of a global prioritized list to local nodes and no updating of a local prioritized list based on the global prioritized list.

The Figure 7 embodiment of the Boland patent provides an alternative system wherein the same single scheduler creates both a global run queue for "non-affined" processes and a plurality of nodal run queues for processes having affinity with the node/processor. The processor, as in Zolnowsky, will look at the global queue and look at the local queue and then select the higher priority process. Once again, it is clear that Boland does not create a global prioritized list, does not maintain or update a local prioritized list, and does not actively communicate any priority information (i.e., a global prioritized list) to the local nodes for use in updating a local prioritized list.

While Custer and the AAPA are again cited in combination with the primary reference, Applicants maintain the position, as set forth above, that neither the Custer nor the AAPA references provide the teachings which are missing from the primary reference. Even if one were to combine the Custer scheduling of more than one application's tasks and the AAPA assignment of same level priorities to all tasks of an application with the Boland teachings, one would not arrive at the present invention since there is no

teaching or suggestion of the claim features of a global scheduler for creating and communicating a global prioritized schedule to local schedulers at which local prioritized schedules are updated. Clearly, therefore, Claims 1, 11, 12 and 14 are not obviated by the combination of the cited teachings.

The Examiner goes on to reject Claims 2, 4-10, 13-16 and 18 by adding the Cameron patent teachings to the combination of Boland, Custer and the AAPA. Applicants rely on the above-stated arguments that the Boland, Custer and AAPA references do not obviate the language of the independent claims 1 and 11, from which Claims 2, 4-10, 13-16 and 18 depend. Moreover, the addition of the Cameron patent does not supply the teachings or suggestions which are missing from the earlier-stated combination. The Cameron patent provides parallel tasks but single level global scheduling with no means for deciding what process or task should execute when a single process of the currently-scheduled parallel job is suspended or waiting. Since all actions are initiated from the Cameron central dispatcher, multiple applications can be assigned to a single processor, however, only one can be active and ready

to run at a time. (Applicants reiterate and once again direct the Examiner's attention to the statement in Col. 4, line 5 et seq of Cameron that "although more than one application is assigned in a partition, an entire application is scheduled at once across all the nodes on which it is loaded"). Those applications are assigned by the central dispatcher for the partition. The Cameron global scheduler issues a single directive to execute a task at a particular time and, again, has no capability to prioritize tasks or to dynamically assign tasks of multiple processes in order of importance to utilize idle CPU time. Clearly, therefore, the Cameron patent does not supply the missing teachings to obviate the invention as claimed.

Finally, with respect to Claims 3 and 17, the Examiner has again cited the combined teachings of Boland, Custer, AAPA, Cameron, and further cites the Ripps reference. As Applicants had previously argued, the Ripps reference simply provides isolated teachings regarding an operating system functionality. Clearly, the addition of the Ripps teachings does not provide the instruction or suggestion to arrive at the invention as claimed, since Ripps also does not teach or suggest the use of a global scheduler for creating and

communicating a global prioritized schedule to local schedulers at which local prioritized schedules are updated

Applicants believe, that, in light of the foregoing explanation and amendments, the currently-pending claims are allowable over the cited art. Withdrawal of the rejections and issuance of the claims it, therefore, respectfully requested.

Respectfully submitted,

M.A. COHEN, ET AL

By: Anne Vachon Dougherty

Anne Vachon Dougherty
Registration No. 30,374
Tel. (914) 962-5910